Dt:- 01/12/25
Day - 06

Loops :-

A loop is used to perform repetitive tasks without writing the same code again and again.

Eg:- If you want to print number 1 to 100, instead of writing 100 print statements, you use a loop.

In Python, two main types of loops.

- for loop - repeats a block of code for each item in a sequ
- while loop - repeats as long as a condition is True.

• nested loops. - loop inside for loop

1. for loop \n"

used when you want to repeat code a specific number of times & iterate over items in a sequence (list, tuple, string etc.)

* used when we know number of interactions.

, Syntax:-

```
for variable in sequence:
    # code
```

example::

```
name = "ajith"
for char in name:
    print (char)
```

Accessing characters using indexing.

```
name = "ragnar"          O/P = r
print (name[0])
```

using enumerate()

```
name = "messi"
for char in enumerate (name):
    print (char)
```

O/P:- (0, 'm')
(1, 'e')
(2, 's')
(3, 's')
(4, 'i')

for loop with range()

Syntax:-

range (start, stop, step)

ex:- print 'Python' 4 times

```
for i in range (1, 8, 1)
    print ("Python")
```

multiplication table Program.

Syntax: n = int (input ("enter a value"))

```
for i in range (1, 11):
    print (f"{n} x {i} = {n*i}")
```

Prime Number check:-

```
num = int (input())
count = 0
for i in range (2, num+1):
    if num % i == 0:
        count += 1
```

➡ loop control statements.

Break - stops the loop immediately.

```
for i in range (1, 11):
    if i == 5:
        break
    print (i)
```

o/p = 1 2 3 4

continue

pass.

Prime Number using for-else.

```
for i in range (2, num):
    if (num % i == 0):
        print ("not prime")
        break
else:
    print ("prime")
```

Counting Natural, even, odd numbers:

```
stv = int()  spv = int()  nn = 0  en = 0  od = 0
for i in range (stv, spv+1):
    nn += 1
    if (i % 2 == 0):
        en += 1
    else:
        od += 1
print (nn)
print (en)
print (od)
```

factorial Program:-

```
num = int(input())

fact = 1
for i in range(1, num+1):
    fact *= i
print(fact)
```